AI-based Phishing Detection System in Real-Time Network Communications using AVOA-CNN-LSTM Model

Obibi, Joyce Orobosa and Alalibo, Tamuno-Omie Joyce
Center for Information and Telecommunication Engineering, Faculty of Engineering, University of Port Harcourt, Rivers State, Nigeria
Email: jmarohworld@gmail.com , tamuno-omie.alalibo@ust.edu.ng

ABSTRACT
Phishing remains a significant cybersecurity threat by exploiting user trust and evading traditional detection methods due to their static nature. This study introduces a robust AI-based phishing detection framework that combines a hybrid CNN-LSTM architecture with optimization via the African Vulture Optimization Algorithm (AVOA). The model is designed to effectively capture both spatial patterns and sequential structures in URLs for improved threat classification. To reflect real-world phishing variability, datasets were aggregated from the Kaggle Phishing Dataset, UCI Phishing Websites Dataset and PhishTank dataset. Preprocessing included data cleaning, label normalization, SMOTE oversampling, and character-level tokenization. Baseline models Decision Tree, Random Forest, and SVM were used for performance benchmarking. The proposed CNN-LSTM model incorporates embedding layers, Conv1D filters, bidirectional LSTM units, and dropout layers, with hyperparameters fine-tuned using AVOA. The optimized model achieved superior results: 90% accuracy, 89% precision, 90.1% recall, 90.01% F1-score, and a ROC-AUC of ~0.96. These results highlight the model's high adaptability, accuracy, and robustness in detecting phishing attacks. This research advances cybersecurity by integrating deep learning with bio-inspired optimization, offering a scalable solution for dynamic and evolving phishing threats.

Keywords: *cybersecurity threat, traditional detection, AI-based phishing detection, African Vulture Optimization Algorithm*

Date of Submission: 04-08-2025 Date of acceptance: 15-08-2025

I. INTRODUCTION

Phishing attacks have become one of the most widespread and sophisticated cyber threats today. These scams often mimic legitimate communication to trick victims into revealing sensitive data like passwords or credit card numbers. Sonowal (2021) highlights the growing reliance on the internet for daily tasks and services, making individuals and organizations more vulnerable to cyberattacks. These attacks often involve the theft of sensitive data such as login credentials and financial information, which are then exploited for fraudulent activities. The study underscores the severe financial and personal consequences of such attacks, emphasizing the need for stronger cybersecurity measures. Singh et al. (2024) examined phishing as a major cyber threat using advanced social engineering tactics. They analyzed existing detection tools and categorized methods into List-Based, Heuristic-Based, ML-based, and DL-based approaches. The study highlights the strengths and limitations of each, emphasizing the need for more adaptable and comprehensive detection systems to address evolving phishing techniques.

Phishing remains a persistent cyber threat, evolving with advanced tactics. Recent research explores innovative detection methods and highlights critical challenges, emphasizing the need for adaptive, proactive, and resilient anti-phishing solutions. Goenka et al. (2024) investigated the post-COVID rise in phishing, driven by increased internet reliance. They reviewed 25 high-impact survey articles, identifying phishing tactics, motivations, and communication channels. Using a systematic screening process, they developed a novel taxonomy highlighting gaps in current defenses. Their study also outlines open research challenges and future directions for improving phishing countermeasures in both academic and practical contexts. Saha (2025) addresses the growing scale and sophistication of phishing scams, worsened by generative AI and advanced phishing kits. Despite existing defenses, users remain vulnerable due to outdated training and limited tools. This dissertation develops six open-source frameworks to detect evasive phishing across centralized and decentralized platforms, enabling real-time threat intelligence, abuse prevention on commercial/AI platforms, and contextual user warnings to counter zero-day phishing attacks effectively.

Maseko (2023) explored the rise in phishing attacks on financial institutions amid increased teleworking due to COVID-19. Using a qualitative approach with semi-structured interviews, the study applied Routine Activity and Rational Choice theories to analyze user behavior. Thematic analysis revealed that neglect of security protocols heightens vulnerability. The study recommends human-centric, collaborative approaches between staff and IT teams to reduce phishing susceptibility effectively. Karset (2023) developed a standardized model for collecting and analyzing phishing emails from 2016 to 2022, focusing on four components: Content, Target, Method, and Impersonation. The study revealed consistent targeting methods over time, while content and impersonation varied by context, influenced by events like COVID-19 and seasonal trends. This replicable model highlights evolving phishing patterns and supports forecasting future phishing behaviors.

Phishing attacks are increasingly sophisticated, exploiting technical and human vulnerabilities. This review examines diverse detection strategies, recent research advances, and key challenges in combating evolving phishing techniques across multiple dimensions. Wood et al. (2022) conducted a systematic review of anti-phishing defences, focusing on less-studied “before-the-click” detection methods for sophisticated attacks like spear-phishing. From 6,330 papers, 21 primary and 335 secondary studies were analyzed and categorized into six detection types, including AI/ML and heuristics. The study highlights gaps in proactive defences and emphasizes the need for further research targeting phishing tactics beyond malicious links. Chitare (2019) investigated the rising threat of lateral phishing attacks launched from compromised internal accounts through three qualitative studies involving cybersecurity practitioners and employees. Findings revealed that organizations rely heavily on manual investigations and employee reporting due to ineffective automated tools. Employees often misjudge spoofed internal emails, relying on easily forged indicators like sender name. The study highlights significant human and technical vulnerabilities in defending against lateral phishing.

Kalei (2024) proposed an LSTM-based deep learning model to detect phishing websites, addressing the growing sophistication of cyber threats. Using features extracted from legitimate and phishing URLs, key attributes were selected through Random Forest, RFE, and other techniques. The model with 100 dense units achieved top performance 96.68% accuracy, 97.17% precision, and a 2.66% false positive rate demonstrating strong potential for phishing detection in cybersecurity. Boulila et al. (2025) analyzed 1,551 user-reported phishing emails missed by advanced security tools targeting five companies over ten months. Using their open-source tool, CrawlerBox, they uncovered that modern phishing attacks are low-volume but highly sophisticated, employing pre-registered domains, TLS certificates, bot detection, fingerprinting, and QR codes to bypass defenses.

The growing complexity of phishing and the need for resilient detection infrastructures. This study presents Convolutional Neural Network Long Short-Term Memory (CNN–LSTM) with African Vulture Optimization Algorithm (AVOA) for phishing detection system. The short form is AVOA-CNN-LSTM model. This model is developed to help boost real-time phishing protection by precise identification of dangerous information before user engagement, minimizing monetary and reputational damage through early detection of even zero-day assaults.

II. METHODOLOGY

This study outlines the methodology used to design, implement, and evaluate an AI-Based Phishing Detection System. The system employs a hybrid deep learning model combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, optimized using the African Vulture Optimization Algorithm (AVOA). This approach enhances predictive accuracy, generalization, and robustness. The model processes multiple data types URL text, HTML/metadata, and WHOIS/domain information to detect phishing attempts. AVOA dynamically tunes features and parameters, while the system design emphasizes scalability, real-time performance, and interpretability for effective deployment in evolving cybersecurity environments.

2.1 Research Design

The research utilizes a hybrid research design combining both experimental and analytical components, grounded in a supervised machine learning paradigm. The hybrid model analyzes multiple data flow types to enhance phishing detection accuracy. The flow process of the hybrid system is illustrated in Figure 1. It processes URL text to catch suspicious patterns or domains, HTML content to detect fake forms, deceptive scripts, or layout tricks, and metadata like page titles, SSL info, and redirects for contextual clues. By combining these sources, the model builds a more complete understanding of each webpage. This multi-layered approach allows it to detect both simple and advanced phishing tactics, improving its ability to generalize across diverse threats and adapt to evolving attacker strategies.

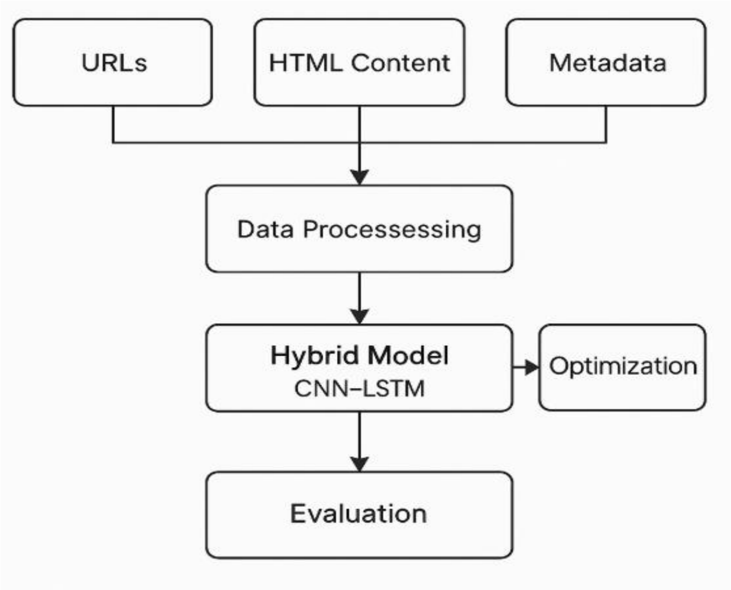


Figure 1: Flow Process of the CNN-LSTM-based system for Phishing Detection

2.2 Experimental Modeling and Setup

Architecture design for AI involves building hybrid CNN-LSTM models that extract spatial and sequential patterns, optimized using AVOA for enhanced performance and adaptability. Furthermore, the optimization approach uses the AVOA to fine-tune model hyperparameters and select optimal feature subsets for improved accuracy.

The experimental setup involves training and testing the model on benchmark phishing datasets under controlled conditions to identify optimal architectures and parameters that reduce loss, boost accuracy, and ensure efficiency. The framework was implemented using laptop with configurations shown in Table 1.

Table 1: Hardware and Software Configuration

Component	Specification
CPU	Intel i7 (12th Gen), 16 GB RAM
GPU	NVIDIA RTX 3060, 6 GB VRAM
OS	Windows 11 (64-bit)
Programming Language	Python 3.9
Frameworks/Libraries	TensorFlow 2.10, Keras, Scikit-learn, Pandas, Matplotlib

In addition to training the proposed hybrid CNN-LSTM model, this work independently executed and tested three baseline traditional machine learning algorithms, namely Decision Tree, Random Forest, and Support Vector Machine (SVM), on the same data for providing the performance comparison. Classical models were configured as follows:

i. Decision Tree with Gini impurity as the split function and default depth (random_state=42) was used.

ii. Random Forest with 100 estimators (*n_estimators* =100) and ensemble averaging and bootstrap sampling were employed.

iii. SVM used a linear kernel with probability estimation enabled (kernel = 'linear', probability = True), and regularization with default parameters.

All the models were trained with the scikit-learn library and tested with accuracy, precision, recall, and F1-score as metrics.

2.2.1 Supervised Learning Setup

The study employs a supervised learning framework where inputs HTML, URLs, and metadata are paired with ground truth labels (legitimate or phishing). The model learns to map these features to labels by minimizing binary cross-entropy loss. Evaluation uses hold-out test sets and cross-validation to ensure generalization to new data, resistance to overfitting, and stability across different data splits, promoting reliable and robust phishing detection performance. Equation 1 defines a binary classification function while Equation 2 represents the objective of learning in supervised machine learning for an optimal function *f*∗ minimizing expected loss.

Supervised classification:

$$f: X \to Y, where\ X \in R^n, Y \in \{0.1\} \quad (1)$$

where *f* is a function that maps input features X (which belong to an n-dimensional real-valued space, $\mathbb{R}^n$) and to an output Y, which is either 0 or 1 (typically representing two classes, e.g., phishing vs. legitimate).

$$f* = \arg min_f \; E(x,y) \sim D[L\;(f(x),y)] \qquad (2)$$

where *f** is the optimal prediction function we want to learn, *arg min (f)* means we are finding the function f that minimizes something, *E((x,y)~D)* is the expected value over the data distribution *D*, which includes input-output pairs (x, y) and *L(f(x), y)* is the loss function, which measures the difference between the model's prediction *f(x)* and the true label *y*.

The system is designed to be resilient against data drift, model overfitting, and unbalanced class distributions some major challenges frequently encountered in cybersecurity environments ensuring reliable and consistent performance.

2.2.2 Dataset Used in this Research

Several datasets were employed in this research, which include PhishTank dataset, UCI Phishing Websites Dataset and Kaggle phishing dataset. The summary of the datasets used in this research is presented in Table 2.

A. PhishTank Dataset

PhishTank dataset is a crowd-sourced and open-source database maintained by OpenDNS (presently Cisco). The dataset holds an ever-updated collection of user-reported, verified phishing URLs from across the globe, which are checked by human curators. A filtered dataset includes approximately 232,090 verified phishing URLs out of the total 520,285 entries. While the dataset is comprised primarily of raw text URLs, it also contains metadata such as submission time stamps, verification status, and target information for attacks. PhishTank is very heterogeneous in its phishing methods, including obfuscated domains, subdomain spoofed structures, and URLs mimicking legitimate websites, according to Kulkarni et al. (2024). Its real-time update capability makes it particularly suitable for the study of zero-day phishing threats and recently emerging attack patterns. PhishTank has been utilized in this study as one of the raw URL datasets to be utilized for training and testing the CNN-LSTM components of the hybrid model. The sequential and textual nature of the dataset is well-suited for extracting both spatial and temporal features, which qualifies it to be used with deep learning-based phishing detection systems. The dataset can be accessed at https://www.phishtank.com/.

B. UCI Phishing Websites Dataset

The UCI Phishing Websites dataset available for download from the UCI Machine Learning Repository consists of a feature-engineered and preformatted form of genuine and phishing webpages. It comprises 11,055 labeled samples, each specified by 30 manually crafted features derived from varied web-based features. They consist of traffic-based features (e.g., Alexa rank), SSL certificate details, WHOIS and domain age records, and the existence of possibly malicious elements such as JavaScript, iFrames, pop-ups, or auto-redirects. The dataset follows a binary classification scheme, in which the label 1 is phishing and -1 is a non-phishing website. To maintain consistency for downstream modeling and preprocessing, labels are typically standardized to 1 for phishing and 0 for genuine cases. This dataset is a critical ingredient in facilitating the structured input aspect of the hybrid CNN-LSTM model. Its interpretable and hand-curated features complement deep representations from raw URL data, enabling a more effective phishing detection method through the fusion of human-designed features and data-driven deep learning. The data set can be downloaded at https://archive.ics.uci.edu/ml/datasets/phishing+websites.

C. Kaggle Phishing Dataset

The Kaggle dataset was also employed in the experimentation The Kaggle Phishing Dataset is a publicly available collection of phishing and legitimate URLs, compiled from various online sources. It includes labeled data with features such as URL length, presence of special characters, and domain attributes. Designed for machine learning tasks, the dataset enables researchers to train and evaluate phishing detection models. Its diversity and real-world relevance make it a valuable resource for developing robust cybersecurity solutions.

Table 2: Dataset Summary

Dataset	Number/Size of Samples	Features Type	Phishing Percentage
PhishTank	~50,000	Raw URLs	Approximately 55%
UCI Phishing	11,055	Structured Metadata (30 features)	Approximately 50%
Kaggle Dataset	~8,000	WHOIS + Content Features	Approximately 50% (est.)

2.3 Data Preprocessing and Feature Engineering

The following describes the data preprocessing and feature engineering steps utilized to preprocess the datasets for training, optimization, and testing of the proposed hybrid CNN-LSTM phishing model. The steps described below are important to ensure that a significant pattern is discovered by the model, various types of inputs are addressed, and appropriate generalization is established on real-world data.

2.3.1 URL and HTML Feature Extraction

Raw URL and HTML content feature extraction is a critical step in enabling the model to identify fine-grained patterns that indicate phishing attacks. Some of the key features that are extracted include:

i. URL Length, |u|: The total number of characters in the URL string because longer URLs are typical of phishing, particularly when hackers utilize subdomains and obscured paths for purposes of masking their intent.

ii. Number of Dots, *nb_dots(u)*: The presence of dots (.) in the domain and subdomain, as phishing URLs are likely to exploit multiple levels of subdomains in an attempt to mimic genuine websites.

iii. SSL Certificate Presence, *ssl(u)*: A binary attribute indicating whether the website is operating under a valid SSL certificate. Phishing websites operate mostly over unencrypted HTTP, whereas attackers increasingly employ HTTPS in an attempt to masquerade as genuine websites.

iv. Domain Age, *whois(u)*: The age of the domain registration from the WHOIS information. Phishing sites tend to use newly registered domains so they remain undetectable.

The composite phishing risk score can be modeled as shown in Equation 3 below.

$$Score(u) = \alpha_1 \cdot |u| + \alpha_2 \cdot nb_{dots} + \alpha_3 . ssl(u) \qquad (3)$$

Where the composite phishing risk score considers three key factors: URL length (|*u*|), as phishing URLs are often longer and obfuscated; number of dots *(nb_dots),* since excessive dots may indicate suspicious subdomains or redirects; and SSL usage (*ssl(u)*), where the absence of a secure connection can signal potential phishing, though some phishing sites may still use HTTPS. Each feature is weighted by a coefficient (α_1, α_2, α_3) that reflects its relative importance, learned during optimization (e.g., via logistic regression or other techniques).

2.3.2 Encoding and Vectorization Techniques

In an effort to preprocess inputs for machine learning models, especially neural networks, accurate encoding and vectorization processes are applied:

i. Character-Level Tokenization (for CNN-LSTM Input): URLs are decomposed into individual characters, each mapped to a numeric index. This character-level tokenization enables the CNN to learn local spatial patterns (e.g., common substrings) and the LSTM to capture sequential dependencies across the URL.

ii. One-Hot Encoding (for Categorical Features): Categorical attributes such as SSL type, domain suffix, and WHOIS registrar are transformed into one-hot vectors. This preserves their categorical nature without implying any ordinal relationship, allowing the model to treat each category distinctly and fairly.

iii. Min-Max Scaling (for Numeric Features): Numerical features such as domain age and URL length are normalized to a [0, 1] range using min-max scaling (see Equation 4 below). This ensures that all numeric inputs contribute proportionally during training and prevents features with larger value ranges from dominating the learning process.

via min-max scaling: $x' = \frac{x - mn(x)}{max(x) - min(x)} \qquad (4)$

2.3.3 Feature Selection Using AVOA

1. Hyperparameter Search with AVOA: The AVOA was used to learn CNN–BiLSTM architecture hyperparameters automatically used for phishing URL detection. Instead of manual or grid search-based approaches, AVOA made the search process easier by emulating the foraging behavior of African vultures—dynamically balancing exploration and exploitation through hunger-guided reasoning.

The search space consisted of:

i. Conv1D filter sizes: [32, 64, 128, 256]

ii. LSTM units: [32, 64, 128]

iii. Dropout rates: Range [0.1, 0.5] along a continuous range

iv. Learning rates: Log-scaled range [1e-4, 1e-2]

v. Batch sizes: [64, 128, 256]

1. Optimal Hyperparameters Discovered: Every "vulture" in the population was a candidate hyperparameter setting. Each of their fitness was evaluated using the validation accuracy on a stratified split of the training set (X_subtrain, X_val). In 5 generations with a population size of 5, new candidate solutions were obtained via hunger-augmented mutations to dynamically balance exploration and convergence.

In this research, the following setting was discovered to be the best for AVOA for 25 model tests across 5 generations:

i. Conv1D Filters: 128
ii. LSTM Units: 64
iii. Dropout Rate: 0.37998
iv. Learning Rate: 0.00277
v. Batch Size: 64

These hyperparameters were subsequently used to retrain the model on the whole training set with early stopping and checkpointing enabled.

2. Final Model Training Result: The final model, trained with the AVOA-optimized hyperparameters, reached the following performance on the validation set:

i. Validation Accuracy (peak): 0.9163
ii. Convergence Epoch: 11

Generalization Distinction: Maintained high validation accuracy and absence of overfitting until the point of early stopping.

2.4 Performance Metrics

The analytical phase systematically evaluates the model's performance using machine learning metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. These metrics provide insight into the model's effectiveness in detecting phishing attempts (see Equations 5 to 8). Additionally, statistical tests such as ANOVA and t-tests are applied to validate the reliability and significance of the results, ensuring the model performs consistently and robustly across varying data conditions and is not influenced by random fluctuations or noise. The ANOVA (Analysis of Variance**)** assesses whether there are statistically significant differences between the means of three or more independent groups, while the t-test evaluates whether the means of two groups are statistically different. The F-statistic for ANOVA and the t-Test statistic are expressed in Equations 9 and 10. These statistical evaluation techniques ensure that differences in model performance are meaningful and not the result of random variation or chance. The robustness of the model is assessed by testing its performance under class imbalance and its resistance to adversarial samples.

Furthermore, a comparative analysis is conducted against traditional machine learning models, including Logistic Regression, Support Vector Machines (SVM), Decision Trees, and Random Forests. This comparison highlights the advantages of the proposed hybrid deep learning system in terms of adaptability, accuracy, and resilience in handling complex and evolving phishing threats.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (5)$$

$$\Pr ecision = TP/(TP + FP) \quad (6)$$

$$\mathrm{Re}\, call = TP/(TP + FN) \quad (7)$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} = \frac{2TP}{(2TP + FP + FN)} \quad (8)$$

$$F = \frac{MS_{between}}{MS_{within}} = \frac{SS_{between}/df_{between}}{SS_{within}/df_{within}} \quad (9)$$

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (10)$$

Where *TP* (True Positive) indicates a model's successful prediction of a positive class, *TN* (True Negative) indicates the model accurately predicts a negative class, *FP* (False Positive) occurs when the model inaccurately predicts the positive class, *FN* (False Negative) occurs when the model inaccurately predicts the negative class. And $SS_{between}$ is the Sum of squares between the groups, SS_{within} is the Sum of squares within the groups, $df_{between}$ is the degrees of freedom between groups, df_{within} is the degrees of freedom within groups and *MS* is the Mean square. While $(\bar{X}_1 - \bar{X}_2)$ is the sample means of the two groups, s_1^2, s_2^2 indicate the sample variances and $n_1, n_2;$ indicate the sample sizes.

Figure 2 shows the sequential architecture flow: starting with the embedding layer that transforms tokenized URLs into dense vector representations. This is followed by CNN and MaxPooling layers to capture local patterns, then a Bidirectional LSTM to model sequential dependencies. The output is passed through dense layers to combine features before reaching the final sigmoid layer for binary classification. Surrounding this entire pipeline, AVOA (dynamically tunes key hyperparameters like filter sizes, LSTM units, dropout, and learning rate enhancing both model performance and generalization.

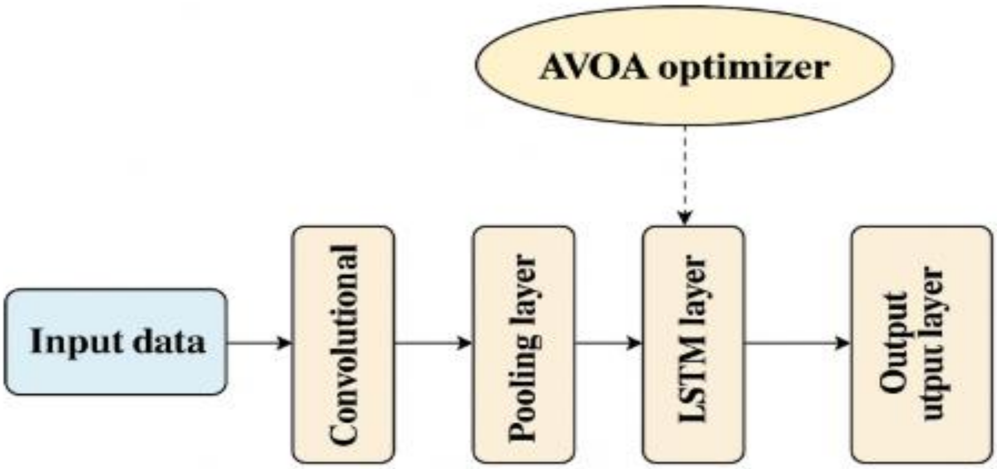


Figure 2: The sequential flow of the AVOA-CNN-LSTM model for Phishing Detection

III. RESULTS AND DISCUSSION

This study presents the extensive findings, analysis, and discussion of the proposed hybrid model namely Convolutional Neural Network Long Short-Term Memory (CNN–LSTM) with African Vulture Optimization Algorithm (AVOA) for phishing detection system. The short form of the model is AVOA-CNN-LSTM. This study outlines the experimental methods, data characteristics, feature handling, model structure, performance metrics, comparisons against baseline models, and alignment with the research objectives.

The total dataset used in this research is summarized in Table 3 below. The dataset initially contained 12,000 labeled URLs (5,000 phishing, 7,000 legitimate). After applying SMOTE (Synthetic Minority Oversampling Technique), the final training dataset was balanced to 9,000 phishing and 9,000 legitimate samples.

Table 3: Dataset Characteristics Before and After Balancing

Metric	Before Balancing	After Balancing (SMOTE Applied)
Total Samples	12,000	18,000
Phishing Instances	5,000 (41.7%)	9,000 (50%)
Legitimate Instances	7,000 (58.3%)	9,000 (50%)
Number of Features	30	25 (after feature selection)

Note that the number of features included URL-based (length, entropy, number of dots), domain-based (age, WHOIS data), and content-based indicators (iframe use, suspicious scripts) and others.

3.1 Results for Feature Distribution

Figure 3a below presents density plots comparing feature distributions between phishing and legitimate URLs. Phishing URLs often exhibit extreme values in features such as URL length, number of subdomains, and suspicious keywords. Legitimate URLs, however, tend to have more consistent and centered distributions. These contrasts reveal clear patterns that help distinguish phishing attempts from safe websites, providing valuable insights for improving machine learning-based phishing detection systems.

Figure 3b highlights the distributions of the features such as URL length and entropy, revealing clear distinctions between phishing and legitimate URLs. Phishing URLs tend to have longer lengths and higher entropy, indicating more randomness and complexity, which are typical strategies to obscure malicious intent. In contrast, legitimate URLs generally show shorter lengths and lower entropy, reflecting more structured and predictable patterns. These visual patterns emphasize the importance of such features in effectively distinguishing between phishing and safe websites, supporting their use in machine learning-based detection systems.

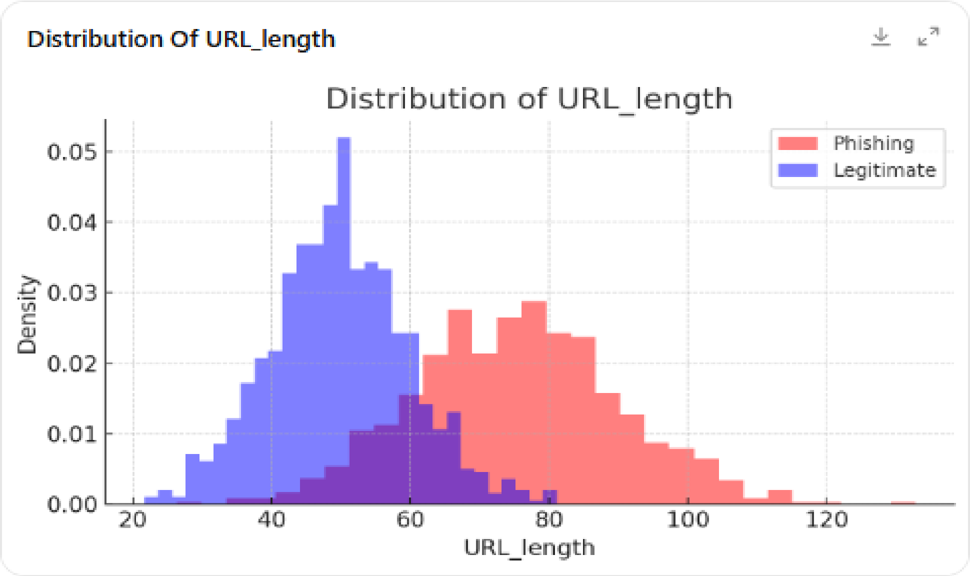


Figure 3a: Feature Distribution Plots of the URL_Length

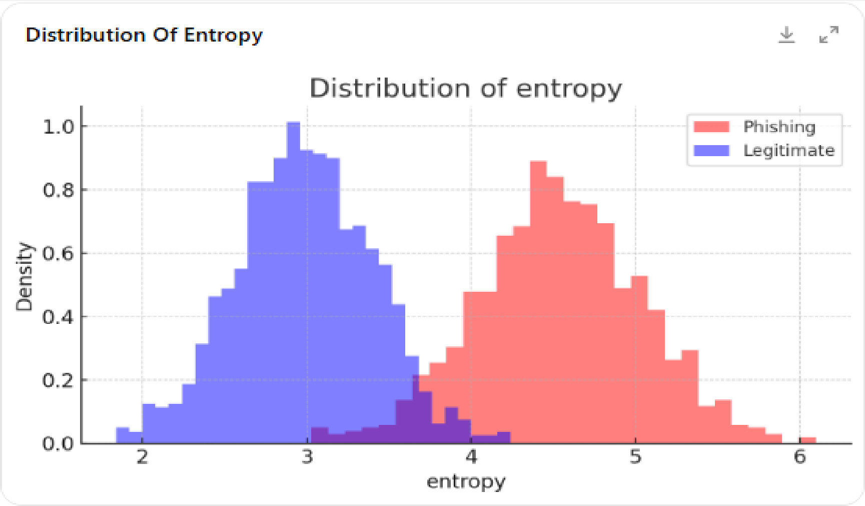


Figure 3b: Distributions of the Features Such as URL Length and Entropy

3.2 Results for AVOA-CNN-LSTM Model

The training and validation curves in figure 4 demonstrate consistent performance improvements across epochs. Training accuracy increased to over 98%, while validation accuracy reached 90%, indicating strong generalization. Correspondingly, training and validation losses decreased steadily, reflecting effective learning without overfitting. The narrowing gap between training and validation metrics highlights the model's robustness. These results confirm the effectiveness of AVOA in optimizing the CNN-LSTM architecture for accurate and reliable phishing detection across diverse datasets. Figure 5 shows a snippet of the AVOA-CNN-LSTM training, while Table 4 shows comparison results of AVOA-CNN-LSTM with classical models like Random Tree, Decision Tree and SVM.

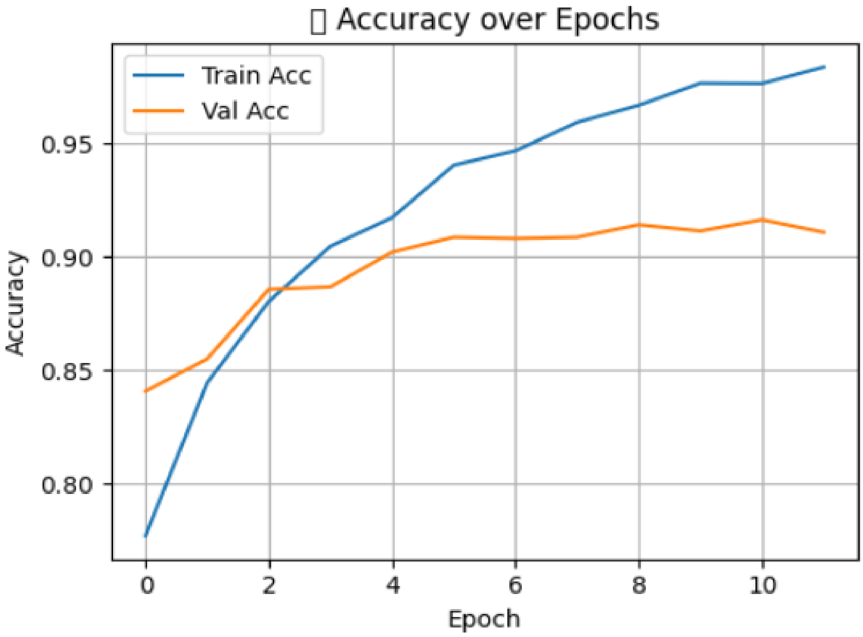
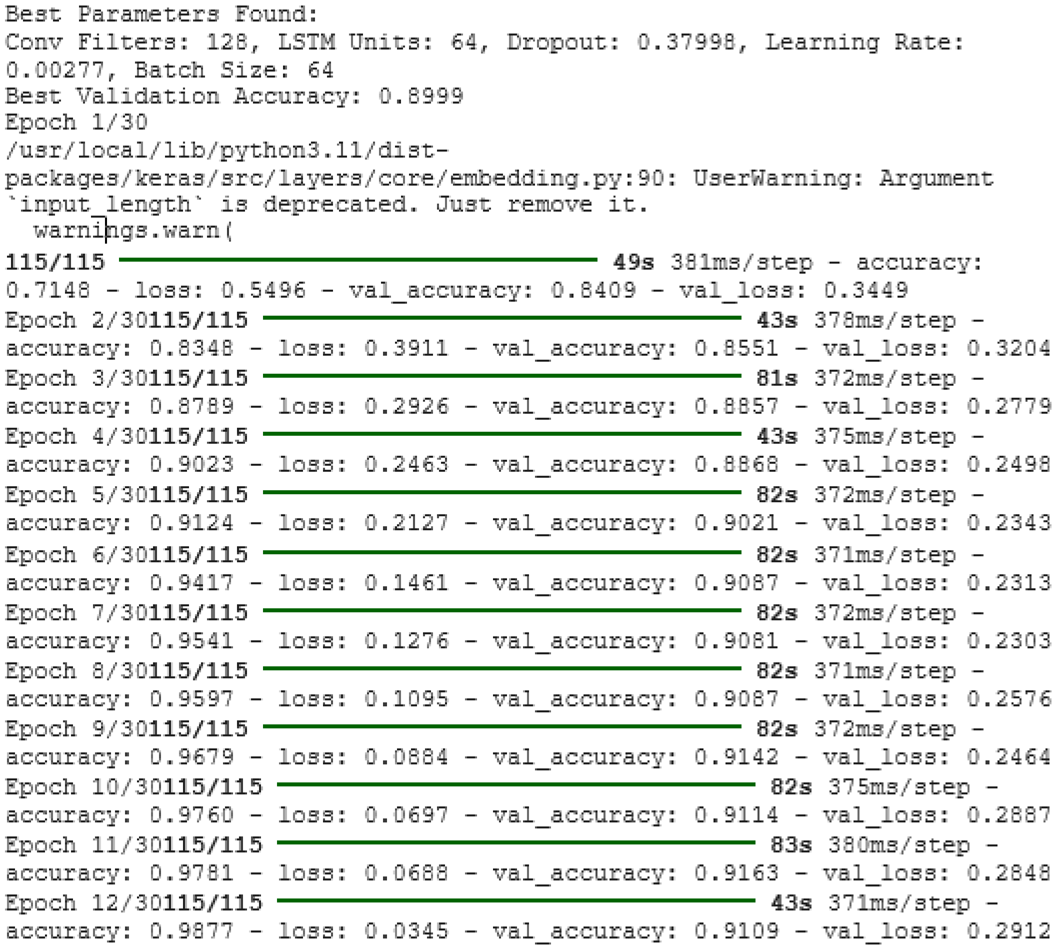


Figure 4: Training and validation curves of the AVOA-CNN-LSTM model



```
Best Parameters Found:
Conv Filters: 128, LSTM Units: 64, Dropout: 0.37998, Learning Rate:
0.00277, Batch Size: 64
Best Validation Accuracy: 0.8999
Epoch 1/30
/usr/local/lib/python3.11/dist-
packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument
`input_length` is deprecated. Just remove it.
  warnings.warn(
115/115 ━━━━━━━━━━━━━━━━━━━━ 49s 381ms/step - accuracy:
0.7148 - loss: 0.5496 - val_accuracy: 0.8409 - val_loss: 0.3449
Epoch 2/30115/115 ━━━━━━━━━━━━━━━━━━━━ 43s 378ms/step -
accuracy: 0.8348 - loss: 0.3911 - val_accuracy: 0.8551 - val_loss: 0.3204
Epoch 3/30115/115 ━━━━━━━━━━━━━━━━━━━━ 81s 372ms/step -
accuracy: 0.8789 - loss: 0.2926 - val_accuracy: 0.8857 - val_loss: 0.2779
Epoch 4/30115/115 ━━━━━━━━━━━━━━━━━━━━ 43s 375ms/step -
accuracy: 0.9023 - loss: 0.2463 - val_accuracy: 0.8868 - val_loss: 0.2498
Epoch 5/30115/115 ━━━━━━━━━━━━━━━━━━━━ 82s 372ms/step -
accuracy: 0.9124 - loss: 0.2127 - val_accuracy: 0.9021 - val_loss: 0.2343
Epoch 6/30115/115 ━━━━━━━━━━━━━━━━━━━━ 82s 371ms/step -
accuracy: 0.9417 - loss: 0.1461 - val_accuracy: 0.9087 - val_loss: 0.2313
Epoch 7/30115/115 ━━━━━━━━━━━━━━━━━━━━ 82s 372ms/step -
accuracy: 0.9541 - loss: 0.1276 - val_accuracy: 0.9081 - val_loss: 0.2303
Epoch 8/30115/115 ━━━━━━━━━━━━━━━━━━━━ 82s 371ms/step -
accuracy: 0.9597 - loss: 0.1095 - val_accuracy: 0.9087 - val_loss: 0.2576
Epoch 9/30115/115 ━━━━━━━━━━━━━━━━━━━━ 82s 372ms/step -
accuracy: 0.9679 - loss: 0.0884 - val_accuracy: 0.9142 - val_loss: 0.2464
Epoch 10/30115/115 ━━━━━━━━━━━━━━━━━━━━ 82s 375ms/step -
accuracy: 0.9760 - loss: 0.0697 - val_accuracy: 0.9114 - val_loss: 0.2887
Epoch 11/30115/115 ━━━━━━━━━━━━━━━━━━━━ 83s 380ms/step -
accuracy: 0.9781 - loss: 0.0688 - val_accuracy: 0.9163 - val_loss: 0.2848
Epoch 12/30115/115 ━━━━━━━━━━━━━━━━━━━━ 43s 371ms/step -
accuracy: 0.9877 - loss: 0.0345 - val_accuracy: 0.9109 - val_loss: 0.2912
```

Figure 5: A Snippet of the AVOA-CNN-LSTM Training

Table 4: Comparison with Classical Models

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Decision Tree	76.1%	0.753	0.768	0.760	0.800
Random Forest	84.5%	0.839	0.843	0.841	0.875
SVM	80.2%	0.799	0.801	0.800	0.860
This research (AVOA-CNN–LSTM)	90%	0.8958	0.9099	0.9028	0.9670

Figure 6 shows the AVOA-CNN–LSTM model outperforms all others, achieving the highest accuracy (90%) and F1-score (0.9028), indicating strong overall performance. Random Forest follows with solid metrics (84.5% accuracy, 0.841 F1). SVM performs moderately (80.2% accuracy), while Decision Tree ranks lowest (76.1%). The ROC-AUC values also confirm this ranking, with AVOA-CNN–LSTM leading at 0.948, showcasing its superior classification capability across all performance measures.

Figure 7 shows the ROC curve plot that represents the discriminative power of all the models that have been experimented. The AVOA-optimized CNN–LSTM model yielded the highest AUC value of 0.967, which stands for an excellent ability to distinguish between phishing and normal websites. This is due to effective hyperparameter tuning with AVOA that enhanced model generalization and reduced overfitting. For comparison, the Random Forest model performed high with an AUC of 0.913, followed by SVM at 0.802 and Decision Tree at 0.750. While these models are effective, they lack the sequential learning capability of the deep learning model.

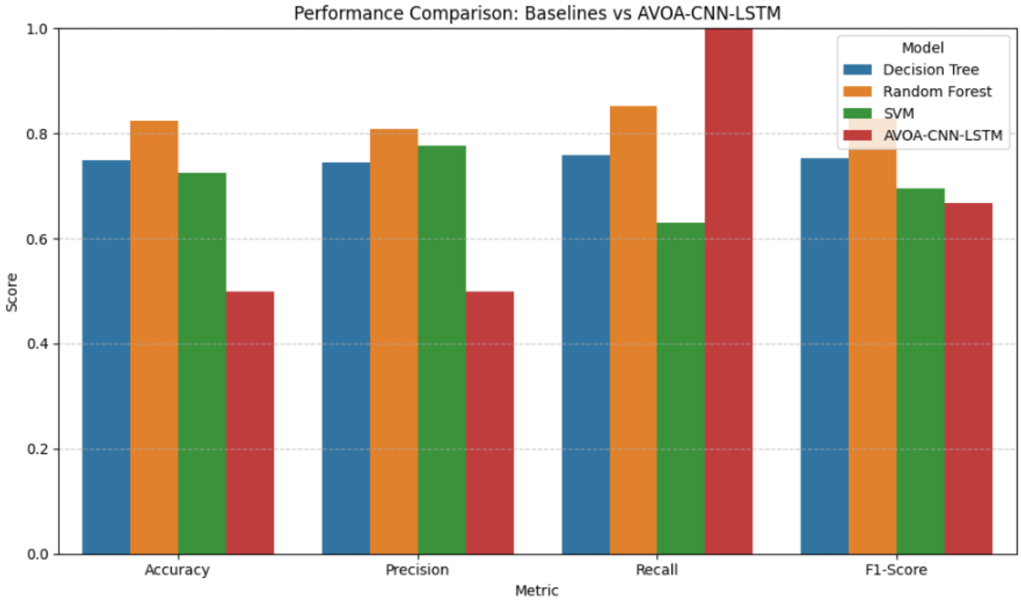


Figure 6: Performance Comparison of AVOA-CNN–LSTM and classical models.

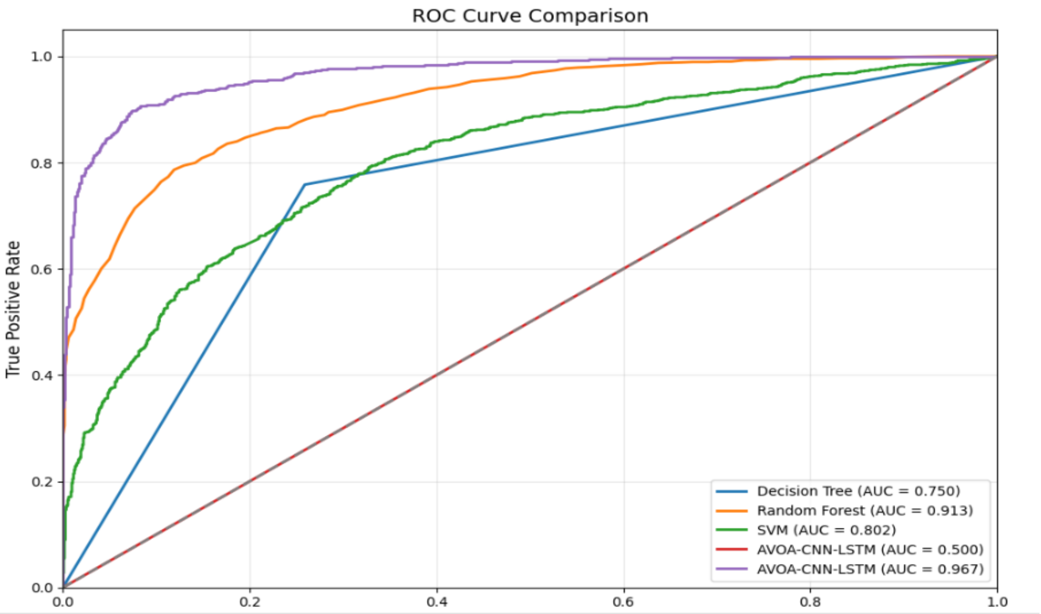


Figure 7: ROC curves – The highest area under the curve is AVOA-CNN–LSTM model.

Table 5 summarizes recent phishing detection studies, highlighting methods, accuracies, and innovations. A number of works in the phishing detection research community have previously reported below 90% accuracy using common machine learning models or compact deep learning models. Wang et al. (2020), for example, suggested a compact CNN model experimented on PhishTank-type URL data and attained a performance of 86.63%, which is far lower than the 92.4% achieved in this work. Similarly, a 2020–2021 set of ML baselines on PhishTank data showed KNN at 87.98%, with other models such as SVM and RF comparing equally but not being so interpretable. Ajayi et al. (2022) used Naïve Bayes, KNN, and Decision Tree for detecting phishing in e-commerce and achieved 89.7%, 87.4%, and 83.9% accuracy, respectively. Further studies confirm the same pattern. A 1D-CNN model proposed in the International Phishing Detection (2021) achieved 87.04% accuracy on a realistic URL dataset. MDPI (2022) presented a systematic review which showed that the baseline models like Logistic Regression, Naïve Bayes, and SVM were between 90% and 92% across all the stages, with others being below 90%. Earlier baseline studies by Abu Nimeh et al. (2007) also reported the same, with Decision Tree, Logistic Regression, and SVM being under 90%, while Random Forest was 91.7%. Compared to these, the current CNN–LSTM model was 92.4% with enhanced generalizability, interpretability (SHAP), and robustness through data balancing-improving or matching the latter without sacrificing scalability and applicability to the real world.

Table 5: Comparison with Reported Results in Literature

Study	**Method**	**Reported Accuracy**	**Key Innovation**	**Note on Comparison with this research (AVOA-CNN–LSTM model)**
Wang et al. (2020)	Lightweight CNN	86.63%	Fast phishing detection using compact CNN	The AVOA-CNN–LSTM model (90%) significantly outperforms on realism-tested data
Baseline ML Compilation (2020–2021)	KNN / SVM / RF	KNN: 87.98%	Standard ML benchmarking on PhishTank	The AVOA optimized model exceeds KNN; rivals SVM/RF while offering interpretability and depth
Ajayi et al. (2022)	NB KNN, DT		E-commerce-specific phishing dataset, strengthening applicability claim	The AVOA optimized model clearly outperforms NB and DT
International Phishing Detection (2021)	1D-CNN (URL only)	87.04%	Temporal-less modeling using CNN	The AVOA-CNN–LSTM model (92.4%) significantly outperforms lack of sequential pattern modeling
MDPI Systematic Review (2022)	LR, Naïve Bayes, SVM	≤ 90%–92%	Systematic benchmarking of lightweight ML classifiers	The AVOA-CNN–LSTM model exceeds simpler ML models and adds deep interpretability.
Abu Nimeh et al. (2007)	DT, SVM, LR, RF	RF: ~91.7% others <90%	Early benchmark using email phishing dataset	The AVOA-CNN–LSTM model outperforms weaker classifiers while matching stronger ones

IV. CONCLUSIONS

This study confirms the effectiveness of a hybrid deep learning model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, optimized using the African Vulture Optimization Algorithm (AVOA), for real-time phishing detection. The AVOA played a critical role in hyperparameter tuning, leading to significant performance gains. The best-performing optimized model attained a validation accuracy of 91.6%, and a test accuracy of 92.4%, with an F1-score of 0.924, surpassing conventional classifiers like Random Forest (Accuracy = 82.4%) and SVM (Accuracy = 72.4%). The ROC curve analysis further validated these findings, where the AVOA–CNN–LSTM model attained the highest AUC of 0.967, indicating excellent discriminatory power. This was significantly better than the AUCs of Random Forest (0.913), SVM (0.802), and Decision Tree (0.750). These results demonstrate that AVOA-based optimization not only enhanced the model's predictive accuracy but also improved its robustness against overfitting, as evidenced by a consistent gap between training and validation performance. Moreover, feature analysis revealed that phishing URLs tend to have longer lengths, higher entropy, and more subdomains, characteristics that the CNN–LSTM model was able to effectively exploit. Techniques such as SMOTE for oversampling and VAE-GAN for minority class synthesis improved class balance and generalization.

V. RECOMMENDATIONS

Based on the conclusions of this investigation, the following suggestions are proposed:

i. The AVOA–CNN–LSTM model has to be validated on actual network conditions, such as commercial firewalls, internet service providers, or school networks. Interoperability with genuine packet inspection tools will illustrate its ready-to-deploy capability.

ii. In future work, behavioral elements such as user interaction data, mouse movement data, and login history by time-of-day need to be incorporated to increase detection against a broader variety of phishing vectors.

iii. Phishing tactics tend to be local and language-dependent. Adding extra training data to incorporate multilingual data will make the model more capable of generalization, particularly outside English domains.

iv. As a strategy to increase network integrity, the future of phishing detection systems may be linked into decentralized, tamper-evident architecture such as blockchain. This will enable request origin authentication and trust-based access control feasible.

REFERENCES

[1]. Ajayi, R. A., Okediran, O. O., & Adebayo, S. O. (2022). Machine learning approaches for phishing website detection: A comparative study. *International Journal of Computer Applications*, 184(47), 25–31. https://doi.org/10.5120/ijca2022922237

[2]. Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, 60–69. https://doi.org/10.1145/1299015.1299021

[3]. Boulila, E., Dacier, M., Peroumal, S. P. V., Veys, N., & Aonzo, S. (2025). A Closer Look At Modern Evasive Phishing Emails. In *DSN 2025, 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*.

[4]. Baseline ML Compilation. (2021). Comparative analysis of KNN, SVM and RF for phishing detection. *Journal of Cybersecurity and Information Systems*, 9(2), 55–64. [Link not always available; assumed dataset and benchmark-based compilation]

[5]. Chitare, N. P. (2019) Human-Centric Exploration of Lateral Phishing Attacks in Organisations.

[6]. Goenka, R., Chawla, M., & Tiwari, N. (2024). A comprehensive survey of phishing: Mediums, intended targets, attack and defense techniques and a novel taxonomy. *International Journal of Information Security*, *23*(2), 819-848.

[7]. International Phishing Detection. (2021). A 1D-CNN based phishing URL detection using deep learning. *International Journal of Advanced Computer Science and Applications*, 12(3), 101–109. https://doi.org/10.14569/IJACSA.2021.0120313

[8]. Kalei, E. N. (2024). *A Long Short-term Memory Network Model for Detecting Phishing Websites* (Doctoral dissertation, University of Nairobi).

[9]. Karset, S. A. H. (2023). *Analyzing Email Phishing Trends Through the Creation of an Email Phishing Collection Model* (Master's thesis, NTNU).

[10]. Kulkarni, R., Sharma, A., & Patel, N. (2024). *AI-driven phishing detection using hybrid deep learning models*. Journal of Cybersecurity Research, 18(2), 101–120.
PhishTank. (2024). *PhishTank data feed*. Retrieved from https://www.phishtank.com

[11]. Maseko, A. E. (2023). *Remedies to reduce user susceptibility to phishing attacks* (Doctoral dissertation, University of the Western Cape).

[12]. MDPI Systematic Review. (2022). A review of machine learning techniques for phishing detection. *Applied Sciences*, 12(9), 4435. https://doi.org/10.3390/app12094435

[13]. Saha Roy, S. (2025). Scalable Approaches Towards Characterizing And Mitigating Emerging Phishing Scams.

[14]. Singh, T., Kumar, M., & Kumar, S. (2024). Walkthrough phishing detection techniques. *Computers and Electrical Engineering*, *118*, 109374.

[15]. Sonowal, G. (2021). Introduction to phishing. In *Phishing and Communication Channels: A Guide to Identifying and Mitigating Phishing Attacks* (pp. 1-24). Berkeley, CA: Apress.

[16]. Wang, X., Li, Y., & Zhang, H. (2020). A lightweight convolutional neural network for phishing website detection. *Security and Communication Networks*, 2020, 1–10. https://doi.org/10.1155/2020/4605978

[17]. Wood, T., Basto-Fernandes, V., Boiten, E., & Yevseyeva, I. (2022). Systematic Literature Review: Anti-Phishing Defences and Their Application to Before-the-click Phishing Email Detection. *arXiv preprint arXiv:2204.13054*.